Git

Otto Sipe

Why Git?

What is a “commit”?

```
git commit -m “did something”
```

What isn’t a “commit”?

```
git commit -m “fu*k code”
git commit -m “mmm. sandwich”
git commit -m “nom. nom.”
```

Say something useful!

First, join a repo.

```
git clone git@domain:user/proj.git
```

(do code)

```
git add .
git commit -m “fixed thingy”
git push
```

(only if you’ve got a remote repo)

(go live life)

```
git pull
```

Pulls other’s changes.
Sync with remote repo.

And that’s it.

Not really!

Opinion:

The point is **not** to know everything about how git works.

Ideally, you don’t care.

If Git were a tool, it’d be a tablesaw.

FELDER

Don’t rush.

Branch

```
git checkout -b branch_name
```

(do work)

```
git commit -am “...”
git push
```

BOOM - new branch.

Why branch?

Don’t mess up everyone else.

Test something new out.

Stable version vs. development.

“master” is default.

Merge

Go to main branch.
Merge in old branch.
Kill old branch.

```
git checkout master
git merge branchname
git branch -d branchname
```

Pro Tips

```
git status
```

What you have changed.

```
git log
```

Show’s commit history.

```
git diff
```

Show changed code.

```
git mv <file> <new file>
```

or

```
git rm <file>
git stash
```

or

```
git stash pop
```

Save all changes for later.

```
git revert <commit>
```

Go back to an old commit.

File: `.gitignore`

What files/ paths to exclude.

File: `README.md`

Merge Conflicts

They’re awful. Terrible.

```
git pull
<horrid merge conflict message and list of files>
vim file.txt
   <<<<<<< HEAD
      // your changes
   =======
      // others changes
   >>>>>>> master
<you need to fix it by hand - pick best code>
git commit -am “fixed merge conflict”
```

Terrible. But fixable.

When you screw up. Google up.

Go forth, and *Git* boldly!

(Breakout Demo)